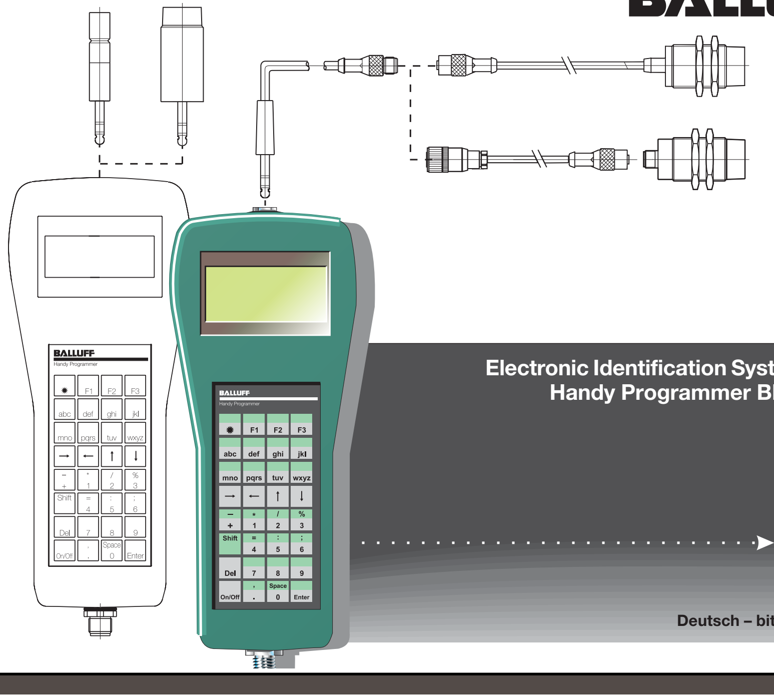
BALLUFF



Manual

Electronic Identification Systems BIS
Handy Programmer BIS C-810

Deutsch – bitte wenden!

Nr. 826 044 D/E • Edition 0510
Subject to modification.
Replaces edition 0410.

Writing convention:

Control characters to be transmitted are in angle brackets.

Characters to be transmitted in ASCII code are enclosed in apostrophes.

Example: <STX> '1 2 3 4 5 6' BCC

http://www.balluff.de

Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Germany
Phone +49 (0) 71 58/1 73-0
Fax +49 (0) 71 58/50 10
E-Mail: balluff@balluff.de

Content

Safety advisory

Approved operation

Series BIS C-810 Handy Programmer along with the other BIS C system components comprise an identification system and may only be used for this purpose in an industrial environment in conformity with Class A of the EMC Law.

Installation and operation

Installation and operation should be carried out only by trained personnel. Unauthorized work and improper use will void the warranty and liability.

When installing the Handy Programmer, follow the chapter “Start-up” closely.

Use and testing

Prevailing safety regulations must be adhered to when using the Identification System. In particular, steps must be taken to ensure that a failure of or defect in the identification system does not result in hazards to persons or equipment.

This includes maintaining the specified ambient conditions and regular testing for functionality of the Identification System including all its associated components.

Function faults

Should there ever be indications that the identification system is not working properly, it should be taken out of commission and secured from unauthorized use.

Scope

This manual applies to Handy Programmer in the series BIS C-810-0-003....

Introduction
BIS C Identification System

This manual is designed to assist the user in setting up the control program and installing and starting up the components of the BIS C Identification System, and to assure rapid, trouble-free operation.

Principles

The BIS C Identification System belongs in the category of

non-contact systems for reading and writing.

This dual function permits applications for not only transporting information in fixed-programmed data carriers, but also for gathering and passing along up-to-date information as well.

Applications

Some of the notable areas of application include

- **for controlling material flow in production processes**
  (e.g. in model-specific processes)**,**
  for workpiece conveying in transfer lines,
  in data gathering for quality assurance ,
  for gathering satety-related data,
- **in tool coding and monitoring;**
- **in equipment organization;**
- **in storage systems for monitoring inventory movement;**
- **in transporting and conveying systems;**
- **in waste management for quantity-based fee assessment.**

Using the portable Identification System with the BIS C-810 Handy Programmer allows data carrier service functions such as checking, correcting or initializing. The data read on the equipment can if needed be stored in the Handy Programmer and transmitted to and then analyzed later on a PC.

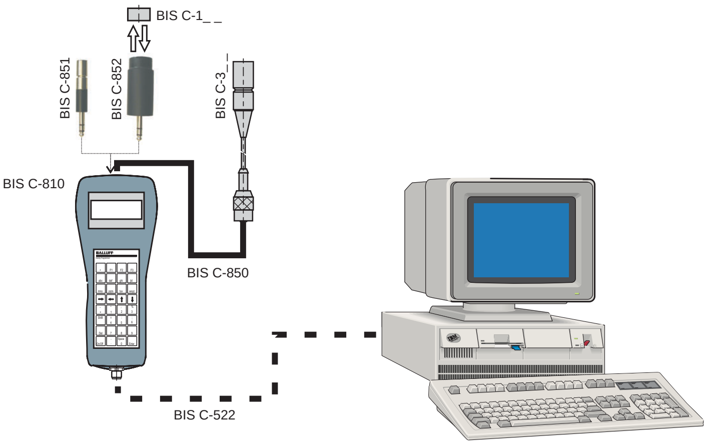
Introduction
BIS C Identification System

System components

The main components of the BIS C Identification System are

- **Handy Programmer,**
- **Read/write heads, and**
- **Data carriers.**

Configuration with Handy Programmer



Schematic representation of an identification system (example)

BIS C-810
Basic knowledge for the application

Operation, keyboard and display

The Handy Programmer features menu-driven operation. Entries are made on a membrane keypad with 32 keys. Some keys have multiple functions. There are two types of multiple function:

2 functions	Switch using the Shift key
3 letters	press 1x = 1st letter, 2× = 2nd letter, 3× = 3rd letter

Switching to capital letters using the Shift key

The LCD display where all data are displayed has 4 lines with 20 characters each.

Language

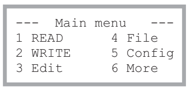
From the main menu you can press Shift+F1 to change the language. The selected language is only retained however until the next time the Handy Programmer is turned off. To make the selection permanent, you must select it in the configuration menu.

Menu/Controls

The main functions of the unit are shown in Main Menu 1. Each sub-menu is indicated by a number. Pressing the appropriate number key opens the corresponding menu and the associated window. In some cases the function is carried out immediately, in other cases you must first make some menu selections.

For quick orientation, the key/key sequence is always indicated in the manual for directly reaching the respective function or menu, always starting from Main Menu 1.

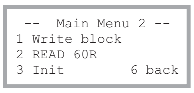
Example: Invoke function "READ 60R", key sequence 6 2



```
---  Main menu   ---
1 READ      4 File
2 WRITE     5 Config
3 Edit      6 More
```

Main menu 1

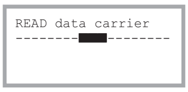
6



```
 --  Main Menu 2 --
1 Write block
2 READ 60R
3 Init        6 back
```

Main menu 2

2



```
READ data carrier
--------███--------
```

Read from data carrier 60R

BIS C-810
Basic knowledge for the application

Battery operation

The device is powered by a rechargeable battery. The current draw when reading or writing is 275 mA. When there is no read/write procedure in process, the unit has a current draw of 65 mA.

The system components are powered by the Handy Programmer. The data carrier represents an independent unit, and does not require any cable-supplied power but rather gets its energy from the read/write head. The latter sends a continuous carrier signal which powers the data carrier as soon as the required distance is reached. The read/write procedure takes place during this phase, and may be static or dynamic.

Selecting read/write heads

Various read/write heads *) are available for the Handy Programmer. These are connected to the Handy Programmer using a jack plug. Which read/write head *) model is used depends on the data carrier type used. The BIS C-850 *) adapter cable can be used to connect a series BIS C-3_ _ read/write head (except BIS C-350 and BIS C-352). For additional information about series BIS C-3_ _ read/write heads including compatibility with various data carriers, please refer to the corresponding manual.

Dialog mode

The Handy Programmer writes to or reads from the data carrier through the read/write head. To prepare data for writing or change data that have been read, the Handy Programmer features an editor function. The data can be read from or written to the data carrier in ASCII, hex, BCD or BINARY format.

Switch formats for the displayed data by pressing the abc and ghi keys:

Key	Format	Key	Format
abc	->ASCII -> BCD -> BINARY	ghi	HEX

*) See Accessories on 🗋 58

BIS C-810
Basic knowledge for the application

Data security for written and read data

When sending data between the read/write head and the data carrier a procedure is required for recognizing whether the data were correctly read or written.

The processor is supplied with standard Balluff procedure of double reading and comparing. In addition to this procedure a second alternative is available: CRC_16 data checking. Here a test code is written to the data carrier, allowing data to be checked for validity at any time or location.

Advantages of CRC_16	Advantages of double reading
Data checking even during the non-active phase (data carrier outside read/write head zone).	No bytes on the data carrier need to be reserved for storing a check code.
Shorter read times since each page is read only once.	Shorter write times since no CRC needs to be written.

Since both variations have their advantages depending on the application, the user is free to select which method of data checking he wishes to use (see Configuration on 🗋 15 "Carrier data").

It is not permitted to operate the system using both check procedures!

BIS C-810
Basic knowledge for the application

Data security for written and read data
(cont.)

To be able to use the CRC check method, the data carriers must be initialized. You either use data crriers with the factory default data set (all data are 0), or you must initialize the data carrier. There are two ways of doing this:

– Initializing the data carrier sing the "Init" function in Main Menu 2 (see 🗋 28).

– Writing to the data carrier using the special initialization command 'Z' (see 🗋 40).

If CRC_16 data checking is activated, a special error message is output to the interface whenever a CRC error is detected.

If the error message is not caused by a failed reed request, it may be assumed that one or more memory cells on the data carrier is defective. That data carrier must then be replaced.

If the CRC error is however due to a failed write request, you must reinitialize the data carrier in order to continue using it.

The checksum is written to the data carrier as a 2-byte wide datum. Two bytes per page are 'lost', i.e., the page size becomes 30 bytes or 62 bytes depending on data carrier type (for setting block size, see 🗋 15 "Selecting data carriers"). This means that the actual usable number of bytes is reduced:

Data carrier type		Usable bytes
128 bytes	=	120 bytes
256 bytes	=	240 bytes
511 bytes *)	=	450 bytes

Data carrier type		Usable bytes
1023 bytes *)	=	930 bytes
2047 bytes *)	=	1922 bytes
2048 bytes	=	1984 bytes

The factory set default setting is for **no** CRC_16 data checking.

*) The last code tag page for these EEPROM-based data carriers is not fully available.

BIS C-810
Basic knowledge for the application

Saving read data

Data which have been read from a data carrier which are now stored in the Handy Programmer can be saved in an EEPROM under a file name of your choice.

Communication between Handy Programmer and PC over the RS 232 interface

The Handy Programmer has its own serial RS 232 port through which read/write jobs can be executed and files exchanged between a PC and the Handy Programmer. For this you will need a BIS C-522 cable *).

Communication on the serial interface uses a specific protocol.

Data transmission between the Handy Programmer and a laptop or PC takes place usine specific telegrams. The following possibilities are offered:

- Using the Handy Programmer to handle read/write jobs from the PC, with the Handy Programmer assuming the function of a BIS Processor. Details see starting 🗋 30.
- File management and data exchange of saved files between the PC and Handy Programmer. Details see starting 🗋 45.

*) See Accessories on 🗋 58

BIS C-810
Start-up

Plugging in the read/ write head

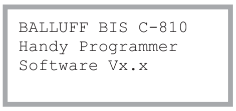
The BIS C-810 Handy Programmer is operated with a read/write head connected to the front side of the device. The read/write head is equipped with a jack plug and is plugged into Terminal X1 on the BIS C-810.

Compatible read/write heads are the BIS C-851, BIS C-852 and BIS C-853.
Alternatively, standard BIS C read/write heads can be used with the BIS C-850 Adapter.

For compatible read/write heads, see Accessories on 🗋 58.

Turning on the Handy Programmer

Turn on the Handy Programmer by pressing the ON button.

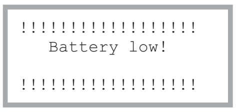


```
BALLUFF BIS C-810
Handy Programmer
Software Vx.x
```

Start display with software version

The unit comes on while displaying the unit type and software version. A battery test is performed at the same time.

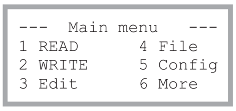
You can shorten the time the display remains on by pressing the Enter key. If the voltage has dropped below a certain nominal capacity, a warning message is issued. The unit can still be operated for a short time in spite of this warning. The battery should however be recharged as soon as possible (see 🗋 54).



```
!!!!!!!!!!!!!!!!!!
   Battery low!

!!!!!!!!!!!!!!!!!!
```

Battery warning message



```
---  Main menu   ---
1 READ      4 File
2 WRITE     5 Config
3 Edit      6 More
```

Main menu 1

After pressing the Enter key the unit indicates its battery state in the following display.

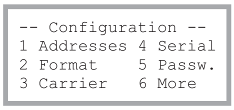
Following a passed battery test this display will appear automatically.

BIS C-810 Configuration

If you do not wish to use the factory setting, start by configuring the Handy Programmer.

Configuration 1
5

Menu item 5 in Main Menu 1 takes you to Configuration Menu 1:



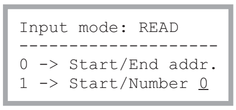
```
-- Configuration --
1 Addresses 4 Serial
2 Format    5 Passw.
3 Carrier   6 More
```

Configuration menu 1

Configuring addresses
5 1

The address ranges for reading and writing of data carriers and for the jump-to address for editing can be preset.

The selection number for the respective mode as well as the desired values must be entered and confirmed by pressing the Enter key. Values you want to copy can be confirmed directly by pressing Enter. In this way you can proceed through the displayed menus one after the other.

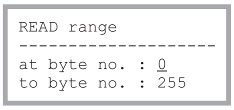


```
Input mode: READ
--------------------
0 -> Start/End addr.
1 -> Start/Number 0
```

Select input mode for reading

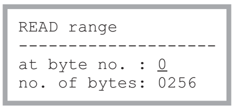
Input mode READ:
Start address / End address or
Start address / Number of bytes for reading

READ range



```
READ range
--------------------
at byte no. : 0
to byte no. : 255
```

Input menu 0 for mode Start/End address for reading



```
READ range
--------------------
at byte no. : 0
no. of bytes: 0256
```

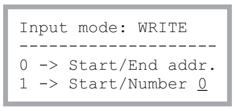
Input menu 1 for mode Start address/No. of bytes for reading

BIS C-810 Configuration

Configuring addresses
5 1
(cont.)

Input mode WRITE

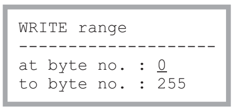
- – Start address / End address or
- – Start address / No. of bytes for writing



```
Input mode: WRITE
--------------------
0 -> Start/End addr.
1 -> Start/Number 0
```

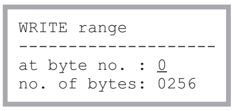
Select input mode for writing

WRITE range



```
WRITE range
--------------------
at byte no. : 0
to byte no. : 255
```

Input menu 0 for mode Start/End address for writing



```
WRITE range
--------------------
at byte no. : 0
no. of bytes: 0256
```

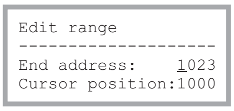
Input menu 1 for mode Start address/No. of bytes for writing

EDIT range

Jump-to address and size for EDIT

Here you must specify:

- – the largest address to be edited, and
- – the address (cursor position) to be displayed first when editing.



```
Edit range
--------------------
End address:    1023
Cursor position:1000
```

Input menu for End address when editing

After the last confirmed entry, the program returns to Main Menu 1.

BIS C-810
Configuration

Select format
5 2

Four data formats are available for entering data in the Editor:

- – hexadecimal (HEX)
- – binary coded decimal (BCD)
- – ASCII
- – BINARY (BIN)

If you choose BCD as the data format, you can also enter a code conversion for the special characters. When BCD is set and data with the numbers A_{Hex} bis F_{Hex} are stored on the data carrier, these numbers are interpreted as special charcters. This function can be used to output BCD values with sign and decimal point. The following special characters are preset:

'A' = * 'B' = + 'C' = , 'D' = – 'E' = . 'F' = /

Select data carrier
5 3

Data carrier specifications ("Carrier data"):

– Capacity:	128, 256, 2048, 8192, 511, 1023, 2047 Byte	
– Block size:	32 bytes	for BIS C-1_ _-02, -03, -04, -05
	64 bytes	for BIS C-1_ _-10, -11, -30
– CRC:	Check	J (Yes), N (No)

Configure port
5 4

Configuring the serial RS 232 port for communicating with a PC or laptop:

The following parameters can be set:

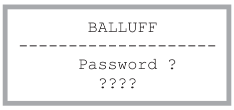
– Baud rate:	9600, 19200, 38400 or 57600 Baud
– Stop bits:	1 or 2
– Data bits:	8 (cannot be changed)
– Parity:	E (even), O (odd) or N (none)
– BCC / CR:	B (BCC), C (CR)

The desired values must be entered and confirmed by pressing Enter. Copied values can be confirmed directly by pressing Enter.

BIS C-810
Configuration

Password
5 5

For security reasons you are prompted for a password when cancelling a write block on the data carrier. The factory set password is 1111.

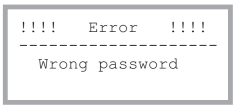


```
      BALLUFF
--------------------
     Password ?
       ????
```

Password entry menu

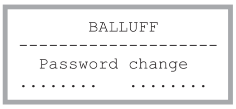
To change the password, you must first enter the current one.

An incorrectly entered password will result in an error message.



```
!!!!   Error   !!!!
--------------------
  Wrong password
```

Error message for incorrect password



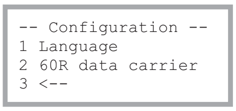
```
      BALLUFF
--------------------
  Password change
........   ........
```

Entry menu for changing the password

After correctly entering the current password, a new 4-digit password may be set.

Configuration 2
Language
5 6 1

Menu item 6 takes you to Configuration Menu 2:



```
-- Configuration --
1 Language
2 60R data carrier
3 <--
```

Configuration menu 2

Here you can select the menu language.

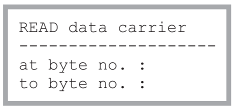
Use menu item 2 to set 60R-001 (1 byte), 60R-002 (8 byte) or 60R-003 (2 byte).

Menu item 3 returns you to Main Menu 1 following a prompt asking whether you want to save the configuration.

Function Description BIS C-810
Processing data carriers

Read data carrier
1

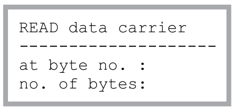
Selecting "READ" in Main Menu 1 takes you to the menu for reading the code tag. You are asked to enter the start address and – depending on which input mode is configured – the end address or the number of bytes:



```
READ data carrier
--------------------
at byte no. :
to byte no. :
```

Entry menu 1 for reading a data carrier

or

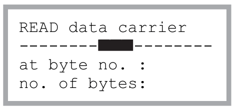


```
READ data carrier
--------------------
at byte no. :
no. of bytes:
```

Entry menu 2 for reading a data carrier

The data for the last entry made or for the data specified in the configuration for start address and end address with number of bytes are displayed as the default values. Press ↓ and ↑ to toggle between the two entry fields. Press Enter to accept the data.

If you press the Enter key to confirm the entered data in the "up to byte no.:" or "number of bytes" field, the read procedure will be started. As long as there is no data carrier in the read range, a bar will flash in the middle of the 2nd line. As soon as a data carrier is recognized, the flashing will stop and the data will be read from the data carrier.



```
READ data carrier
--------████--------
at byte no. :
no. of bytes:
```

Screen shown while reading a data carrier

⚠ The data carrier must not be removed from the active zone of the read/write head while data are being read from the data carrier!

To cancel the read, press the Enter key.

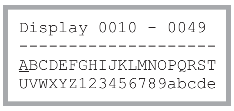
Function Description BIS C-810
Processing data carriers

Read data carrier
1
(cont.)

After a successful read, the bar disappears and the data are shown in the display.

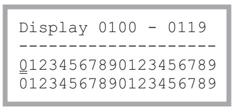
Depending on which format you have selected, the data will be displayed as hex values, ASCII characters, BCD numbers or BINARY. Use the abc and ghi keys to toggle between the various formats:

Key	Format	Key	Format
abc	->ASCII -> BCD -> BIN	ghi	HEX



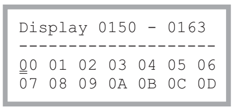
```
Display 0010 - 0049
--------------------
ABCDEFGHIJKLMNOPQRST
UVWXYZ123456789abcde
```

Data displayed in ASCII format



```
Display 0100 - 0119
--------------------
01234567890123456789
01234567890123456789
```

Data displayed in BCD format



```
Display 0150 - 0163
--------------------
00 01 02 03 04 05 06
07 08 09 0A 0B 0C 0D
```

Data displayed in HEX format

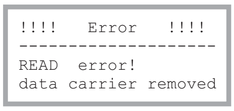
Pressing Enter returns you to Main Menu 1.

Function Description BIS C-810
Processing data carriers

Read errors

If an error occurs while reading, an error message is shown in the display.

Error message "**Data carrier removed**"



```
!!!!   Error   !!!!
--------------------
READ  error!
data carrier removed
```

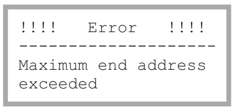
Error message 1 for reading

Causes:

– Data carrier was removed from the active zone too soon
– Data carrier – read/write head distance too great

After acknowledging the error message by pressing Enter you will be returned to Main Menu 1.

Error message "**Maximum end address exceeded**"



```
!!!!   Error   !!!!
--------------------
Maximum end address
exceeded
```

Error message 2 for reading

Causes:

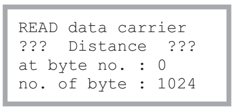
– End address is greater than the configured capacity of the data carrier

After acknowledging the error message by pressing Enter you will be returned to the entry screen for reading.

Function Description BIS C-810
Processing data carriers

Read errors
(cont.)

Error message "**??? Distance ???**"



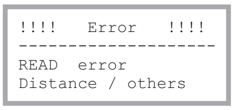
```
READ data carrier
???  Distance  ???
at byte no. : 0
no. of byte : 1024
```

Error message 3 for reading

Causes:

– Data carrier – read/write head distance too great
– End address (= Start address + no. of bytes) is greater than the capacity of the data carrier
– Wrong block size configured for this data carrier
– Data carrier defective

The Handy Programmer repeats the read operation another 2 times. If the 3rd attempt also fails, the operation is cancelled with this error message:



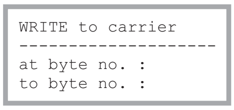
```
!!!!   Error   !!!!
--------------------
READ  error
Distance / others
```

Error message 4 for reading

Function Description BIS C-810
Processing data carriers

Writing to data carriers
[2]

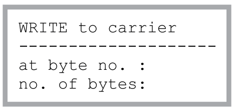
If "WRITE" is selected in the Main Menu, the entry screen for writing to the data carrier is displayed. You are prompted for the start address and – depending on the input mode configured – the end address or number of bytes.



```
WRITE to carrier
--------------------
at byte no. :
to byte no. :
```

Entry menu 1 for writing to the data carrier

or

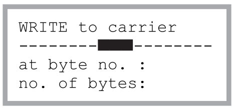


```
WRITE to carrier
--------------------
at byte no. :
no. of bytes:
```

Entry menu 2 for writing to the data carrier

The data for the last entry resp. the data for start address and end address or number of bytes are used as default values. Press [↓] and [↑] to toggle between the two entry fields. Press [Enter] to accept the data.

Pressing the [Enter] key to confirm the data entered in the "to byte no.:" or "number of bytes" field starts the write operation. As long as there is no data carrier in the read range, a bar will flash in the middle of the 2nd line. As soon as a data carrier is recognized, the flashing will stop and the data will be read from the data carrier.



```
WRITE to carrier
--------████--------
at byte no. :
no. of bytes:
```

Screen shown while writing to the data carrier

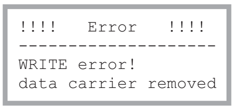
⚠ The data carrier must not be removed from the active zone of the read/write head while data are being read from the data carrier!

Function Description BIS C-810
Processing data carriers

Write errors

If an error occurs while writing, an error message is shown in the display.

Error message "**Data carrier removed**"



```
!!!!   Error   !!!!
--------------------
WRITE error!
data carrier removed
```

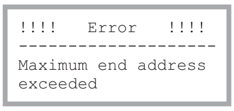
Error message 1 for writing

Causes:

- Data carrier was removed from the active zone too soon
- Data carrier – read/write head distance too great

After acknowledging the error message by pressing [Enter] you will be returned to Main Menu 1.

Error message "**Maximum end address exceeded**"



```
!!!!   Error   !!!!
--------------------
Maximum end address
exceeded
```

Error message 2 for writing

Causes:

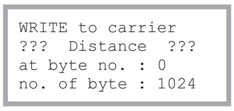
- End address is greater than the configured capacity of the data carrier

After acknowledging the error message by pressing [Enter] you will be returned to the entry screen for writing.

Function Description BIS C-810
Processing data carriers

Write errors
(cont.)

Error message "**??? Distance ???**"



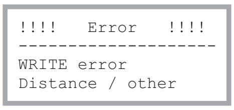
```
WRITE to carrier
???  Distance  ???
at byte no. : 0
no. of byte : 1024
```

Error message 3 for writing

Causes:

- Data carrier – read/write head distance too great
- End address (= Start address + no. of bytes) is greater than the capacity of the data carrier
- Wrong block size configured for this data carrier
- Data carrier defective

The Handy Programmer repeats the write operation another 2 times. If the 3rd attempt also fails, the operation is cancelled with this error message:



```
!!!!   Error   !!!!
--------------------
WRITE error
Distance / other
```

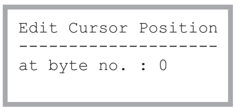
Error message 4 for writing

Function Description BIS C-810
Processing data carriers

Editing data
3

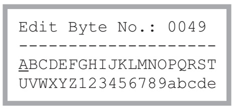
Menu point 3 "Edit" in Main Menu 1 opens the Editor.

In the next display you enter the start address into the Editor (0-8191).



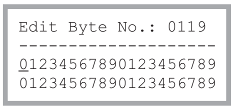
```
Edit Cursor Position
--------------------
at byte no. : 0
```

The data are displayed as HEX values, ASCII characters, BCD numbers or BINARY depending on the format selected. The character highlighted by the cursor can be edited.



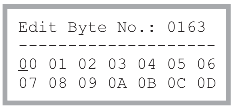
```
Edit Byte No.: 0049
--------------------
ABCDEFGHIJKLMNOPQRST
UVWXYZ123456789abcde
```

Editing data in ASCII format



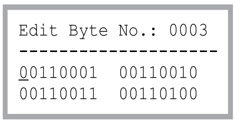
```
Edit Byte No.: 0119
--------------------
01234567890123456789
01234567890123456789
```

Editing data in BCD format



```
Edit Byte No.: 0163
--------------------
00 01 02 03 04 05 06
07 08 09 0A 0B 0C 0D
```

Editing data in HEX format



```
Edit Byte No.: 0003
--------------------
00110001  00110010
00110011  00110100
```

Editing data in BINARY format

Press → and ← to go to the adjacent character. If HEX is set, pressing → and ← advances by a half-byte.

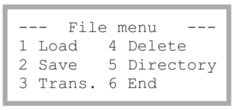
Pressing ↓ uand ↑ jumps one line higher or lower. If the display range is exceeded, the system automatically scrolls up or down by one line.

Press Enter to return to Main Menu 1.

Function Description BIS C-810
Working with files

File menu
4

The BIS C-810 Handy Programmer contains a non-volatile EEPROM memory. This EEPROM can be used to store data read from a data carrier under a file name. The file name is limited to 8 characters. Up to 15 files having a maximum of 2048 bytes per file can be stored.



```
---  File menu   ---
1 Load   4 Delete
2 Save   5 Directory
3 Trans. 6 End
```

File menu

File format

Data carrier files have the following structure:

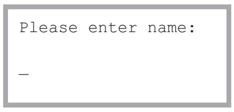
Start address of the data on the data carrier	4 bytes ASCII
No. of bytes	4 bytes ASCII
Data carrier data	Code-transparent; data amount as specified in number of bytes

Example: Data carrier file starting at Address 75 with 16 bytes of data in ASCII format
Data: ABCDEFGHIJKLMNOP
File contents: 00750016ABCDEFGHIJKLMNOP

Load file
4 1

Menu point 1 "Load" in the file menu can be used to load a file stored in the EEPROM into RAM of the Handy Programmer.

The data in this file can then be further processed by selecting menu item 3 "Edit" in Main Menu 1.



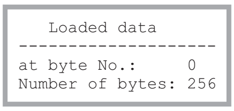
```
Please enter name:

_
```

Load file from EEPROM

After loading the data, the display shows the start address and the end address or number of bytes. Pressing the Enter key qits the display.

The data from this file can then be further processed selecting menu item 3 "Edit" in Main Menu 1.

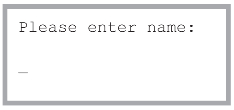


```
   Loaded data
--------------------
at byte No.:     0
Number of bytes: 256
```

Function Description BIS C-810
Working with files

Save file
4 2

Use menu item 2 "Save" to save the data that were last read from a data carrier and still contained in RAM of the Handy Programmer as a file in the EEPROM memory.



```
Please enter name:

_
```

Save file in EEPROM

Exchanging data over the serial interface
4 3

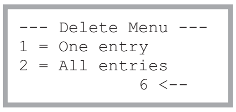
Using menu item 3 "Trans." (for data transmission) you can transmit files back and forth between a PC or laptop to the Handy Programmer over the serial interface. For additional information see 🗋45ff.

Delete file(s)
4 4

Menu point 4 "Delete" allows you to delete saved files.

Menu point 1 deletes a single file. An input screen opens where you enter the file name.

Menu point 2 deletes all saved files at one time.



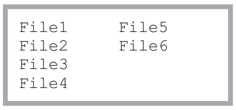
```
--- Delete Menu ---
1 = One entry
2 = All entries
            6 <--
```

Delete menu

Show directory
4 5

Menu point 5 "Directory" displays a lists of the saved files

Press Enter to exit the display.



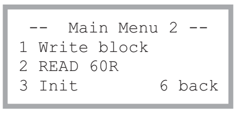
```
File1     File5
File2     File6
File3
File4
```

Directory

Function Description BIS C-810
Special data carrier functions

Main Menu 2
6

Selecting menu point 6 "More" in Main Menu 1 takes you to Main Menu 2.



```
 --  Main Menu 2 --
1 Write block
2 READ 60R
3 Init        6 back
```

Main menu 2

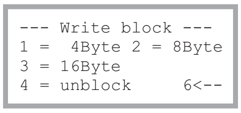
Write block on data carrier
6 1

Some data carriers allow you to block parts of the memory from writing. There are 3 options:

- Data carrier address 0 to 3 (4 bytes)
- Data carrier address 0 to 7 (8 bytes)
- Data carrier address 0 to 15 (16 bytes)

The write block is stored on the data carrier.
It cannot be cancelled using standard Processors.

After selecting menu point 1 "Write block", the "Write block" menu opens.



```
--- Write block ---
1 =  4Byte 2 = 8Byte
3 = 16Byte
4 = unblock     6<--
```

Write block menu

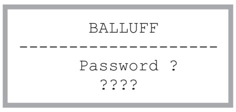
Set write block
6 1 ...

Entering 1, 2 or 3 directly activates the write block for the corresponding number of bytes on the data carrier.

Cancel write block
6 1 4

Entering 4 "Unblock" requires the use of a password.

After entering the correct password, the write block is lifted for the data carrier.

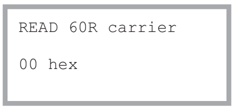


```
       BALLUFF
--------------------
     Password ?
        ????
```

Password entry menu

Function Description BIS C-810
Special data carrier functions

Read 60R
6 2



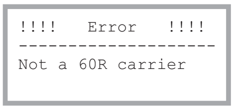
```
READ 60R carrier

00 hex
```

Read 60R data carrier (example)

Menu point 2 "Read 60R" in Main Menu 2 starts the read of a 60R data carrier. After the read operation, the read value is displayed.

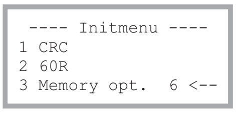
If the data carrier cannot be read, this error message appears:



```
!!!!   Error   !!!!
--------------------
Not a 60R carrier
```

Error message when reading a 60R data carrier

Init menu: Initialize data carrier
6 3



```
 ---- Initmenu ----
1 CRC
2 60R
3 Memory opt.   6 <--
```

Init menu

Selecting menu point 3 "Init" in Main Menu 2 takes you to the Init menu.

CRC

Menu point 1 "CRC" writes 00_{Hex} to the data carrier, which prepares for operation with a CRC data check.

60R

Menu point 2 "60R" opens the input screen for the hex-value number (depending on configuration). When starting your entry the cursor is at Byte 0. After entering the desired Hex value, the data carrier is initialized with this value.

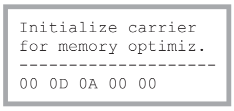
Function Description BIS C-810
Special data carrier functions

Init menu: Initialize data carrier (cont.)

Memory opt.

Menu point 3 "Memory opt." opens the input screen for the 5 initialization bytes. After entering the 5 bytes, the data carrier is ready for operation with memory optimization.

Refer to the information in the respective user's manual for BIS Processors with memory optimization for the meaning of the individual bytes.



```
Initialize carrier
for memory optimiz.
--------------------
00 0D 0A 00 00
```

Initialize data carrier for memory optimization

Function Description BIS C-810 / PC
Reading / Writing over the serial interface

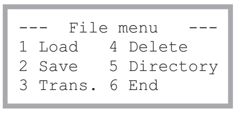
In addition to key-controlled processing of the data carriers, it is also possible to perform read and write operations from a PC or laptop over the serial interface of the Handy Programmer. The Handy Programmer then functions just like a BIS Processor.

The PC and Handy Programmer are connected using a BIS C-522 *) cable. Communication between the Handy Programmer and the laptop/PC uses a specific protocol (see □ 31).

For details on the individual telegrams see □ 32ff.

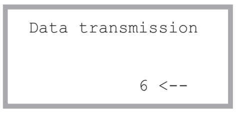
Activate data transmission
4 3

Menu point 4 "File" is selected in Main Menu 1 and menu point 3 "Trans." activated in the following file menu to start this function.



```
---  File menu   ---
1 Load    4 Delete
2 Save    5 Directory
3 Trans.  6 End
```

File menu



```
 Data transmission


            6 <--
```

Data transmission

Data integrity on the RS 232 interface

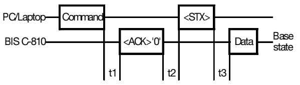
All commands for reading and writing (L, Z, C, P, S and U) as well as the Acknowledge command (Q) can, depending on the configuration, be terminated either with a BCC block check or with Carriage Return 'CR'. The Handy Programmer then also terminates all transmitted characters using the selected termination type.

*) See Accessories on □ 58

Function Description BIS C-810 / PC
Reading / Writing over the serial interface

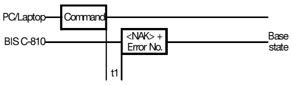
Protocol sequence

Read: a) If no error:



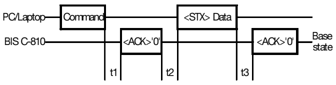
t1 depending on no. of bytes to read (see 🗋 44)
t2 ≥ 0 (is not monitored by the processor)
t3 = max. 50 ms

b) With error:



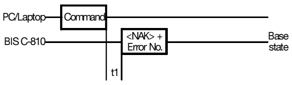
t1 depending on no. of bytes to read (see 🗋 44) and error type (recommended monitor time: 15 s)

Write: a) If no error:



t1 = max. 50 ms
t2 ≥ 0 (is not monitored by the processor)
t3 depending on no. of bytes to write (see 🗋 44)

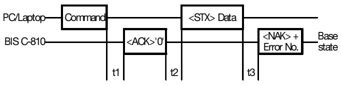
b) With error in command:



t1 = max. 50 ms

The examples are valid only if:
- The Handy Programmer is in the base state.
- A data carrier is present in front of a read/write head.

c) With error in writing:



t1 = max. 50 ms t2 ≥ 0 (is not monitored by the laptop adapter)
t3 depending on no. of bytes to write (see 🗋 44) and error type (recommended monitor time: 30 s for data carriers with 32 byte block size, 60 s for data carriers with 64 byte block size)

Function Description BIS C-810 / PC
Reading / Writing over the serial interface

Telegram types with associated command (ASCII)

There are specific telegrams for the individual operations in the BIS C-810 Handy Programmer. They always begin with the command which corresponds to the telegram type:

'L'	Read data carrier
'P'	Write to data carrier
'C'	Write a constant value to the data carrier
'Q'	Restart Handy Programmer (Quit)
'Z'	Initialize data carrier for CRC_16 data check
'S'	Read a 60R data carrier
'U'	Write a Hex value to the 60R data carrier

Please note:
– Continuous querying on the interface is not permitted!
– The minimum wait time between two commands is 300 ms!

Function Description BIS C-810 / PC
Reading / Writing over the serial interface

Explanation of selected telegram contents

Start address and no. of bytes	The start address (A3, A2, A1, A0) and the number of bytes to send (L3, L2, L1, L0) are transmitted as ASCII characters. For the start address a range of 0000 to 8191 and for the number of bytes 0001 to 8192 can be used. A3 ... L0 stand for 1 ASCII character each. **Please note:** Start address + number of bytes may not exced the data carrier capacity.

Block size	For the 'L', 'P', 'Z', and 'C' commands, the block size B ('0', '1') of the data carrier has to be sent.
	B = '0' corresponds to 64 bytes,
	B = '1' corresponds to 32 bytes.
	Note: Preceding the block size is a '1' for the fixed-set head number 1.

Acknowledgement	The <ACK> '0' is sent by the identification system if the serially transmitted characters were correctly recognized and a data carrier is within the active zone of a read/write head. <NAK> + 'Error No.' is sent as an acknowledgement if an error is detected or if there is no data carrier within the active zone of a read/write head.
Start	<STX> starts data transmission.
Bytes sent	The data are transmitted code-transparent (not converted).

Function Description BIS C-810 / PC
Reading / Writing over the serial interface

BCC Block Check

The BCC block check is formed as an EXOR of the serially transmitted binary characters of the telegram block. Example: Read 128 bytes starting at address 13.
The command line without BCC is: 'L 0013 0128 10'

The BCC is formed:	'L	=	0100 1100	EXOR
	0	=	0011 0000	EXOR
	0	=	0011 0000	EXOR
	1	=	0011 0001	EXOR
	3	=	0011 0011	EXOR
	0	=	0011 0000	EXOR
	1	=	0011 0001	EXOR
	2	=	0011 0010	EXOR
	8	=	0011 1000	EXOR
	1	=	0011 0001	EXOR
	0'	=	0011 0000	EXOR
Block check result:	BCC	=	0100 0100	= 'D'

Variant for terminating with 'CR' instead of BCC

If necessary, termination using block check BCC can be replaced with a special ASCII character. This is:

– Carriage Return 'CR'

Example: Read 128 bytes starting at address 13.

The command line with 'CR" as a terminator is then: 'L 0013 0128 10 CR'

Telegram Structure and Examples
Reading / Writing over the serial interface

Read from data carrier with block size select
Write to data carrier with block size select

Task	Data flow	Com-mand	Start address of first byte to be set	Number of bytes to be sent		Block size	End	Acknow-ledge 2)	Start trans-mission	Data (from start address to start address + no. of bytes)	End	Acknow-ledge 2)
Read	from host system to BIS	**'L'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '8 1 9 2'	'1'	B '0' or '1'	BCC or 'CR'		<STX>			
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.		D1 D2 D3 ... Dn	BCC or 'CR'	
			1)									
Write	from host system to BIS	**'P'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	'1'	B '0' or '1'	BCC or 'CR'		<STX>	D1 D2 D3 ... Dn	BCC or 'CR'	
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.				<ACK>'0' or <NAK> + Error-No.
			1)							1)		

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Telegram Structure and Examples
Reading / Writing over the serial interface

Telegram example for 35:
Read from data carrier with block size select
with BCC block check

You want to read 10 bytes from the data carrier starting at address 50. The data carrier has a block size of 64 bytes.

The host sends 'L 0 0 5 0 0 0 1 0 1 0 I' BCC (4A Hex)

Address of first byte to read
Number of bytes to read
Fixed
Block size 0 = 64 bytes

The Handy Programmer acknowledges with <ACK> '0'
The host gives the start command <STX>
The Handy Programmer provides the data from the data carrier '1 2 3 4 5 6 7 8 9 A F' BCC (70 Hex)

Telegram example for 35:
Write to data carrier with block size select
with BCC block check

You want to write 5 bytes to the data carrier starting at address 500. The data carrier has a block size of 64 bytes.

The host sends 'P 0 5 0 0 0 0 0 5 1 0 Q' BCC (51 Hex)

Address of first byte to write
Number of bytes to write
Fixed
Block size 0 = 64 Byte

The Handy Programmer acknowledges with <ACK> '0'
The host gives the start command and data <STX> '1 2 3 4 5 3' BCC (33 Hex)
The Handy Programmer acknowledges with <ACK> '0'

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Telegram Structure and Examples
Reading / Writing over the serial interface

Writing a constant value in the data carrier with block size select

This command can be used to erase a data carrier data. One saves the time for the transmission of the write byte.

Task	Data Flow	Com-mand	Start address of first byte to be sent	Number of bytes to be sent		Block size	End	Acknow-ledge 2)	Start trans-mission	Data (from start address to start address + no. of bytes)	End	Acknow-ledge 2)
Write	from host system to BIS	**'C'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	'1'	B '0' or '1'	BCC or 'CR'		<STX>	D	BCC or 'CR'	
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.				<ACK>'0' or <NAK> + Error-No.
			1)							1)		

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Telegram Structure and Examples
Reading / Writing over the serial interface

Telegram example for 🗋37:
Writing a constant value in the data carrier with block size select
with BCC block check

Write ASCII data value 0 (30H) to 500 bytes of the data carrier starting a Address 20. The data carrier has a block size of 64 bytes.

The host sends 'C 0 0 2 0 0 5 0 0 1 0 E' BCC (46 Hex)

Address of first byte to write
Number of bytes to write
Fixed
Block size 0 = 64 Byte

The Handy Programmer acknowledges with	<ACK>	'0'
The host gives the start command and data	<STX>	'0 2' BCC (32 Hex)
The Handy Programmer acknowledges with	<ACK>	'0'

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Telegram Structure and Examples
Reading / Writing over the serial interface

Restart the Handy Programmer (Quit)

Sending the Restart command causes a telegram in process to be aborted and puts the Handy Programmer in the ground state.
After this telegram is acknowledged, a minimum of 1600 ms pause should be allowed before starting a new telegram.

☞ Important! The Quit command is not permitted while the Handy Programmer is waiting for a terminator (BCC or 'CR'). In this situation, the Quit would be incorrectly interpreted as a terminator or datum.

Task	Data Flow	Command	Terminator	Acknowledge	Terminator
Restart (Quit)	from host system to BIS	**'Q'**	BCC or'CR'		
	from BIS to host system			'Q'	BCC or'CR'
			1)		

1) The Quit command is not permitted at this point.

Telegram example with BCC block check:

Put the Handy Programmer into the ground state.

The host sends	'Q Q'	BCC (51 Hex)
The Handy Programmer acknowledges with	'Q Q'	BCC (51 Hex)

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Telegram Structure and Examples
Reading / Writing over the serial interface

Initialize a data carrier for CRC_16 data checking

This telegram initializes a data carrier located at the active read/write head for use of CRC_16 data checking.
This telegram must also be send again if a CRC error results from a failed write operation, i.e., the data carrier must be reinitialized in order to use it again.

Please note the table on 🗋 10! The indicated number of usable bytes may not be exceeded, i.e., the sum of start address plus number of bytes must not exceed the data carrier memory capacity!

Task	Data Flow	Command	Start address of first byte to be sent	Number of bytes to be sent		Block size	End	Acknowledge 2)	Start transmission	Data (from start address to start address + no. of bytes)	End	Acknowledge 2)
Initialize CRC_16 range	from host system to BIS	**'Z'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	'1'	B '0' or '1'	BCC or 'CR'		<STX>	D1 D2 D3 ... Dn	BCC or 'CR'	
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.				<ACK>'0' or <NAK> + Error-No.
			1)							1)		

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Telegram Structure and Examples
Reading / Writing over the serial interface

Reading a 60R data carrier

This telegram is used to read a 60R data carrier.

Task	Data Flow	Com-mand	End	Acknowledge 2)	Start transmission	Datum	End
Read a 60R data carrier	from host system to BIS	**'S'**	BCC or 'CR'		<STX>		
	from BIS to host system			<ACK>'0' or <NAK> + Error-No.		D	BCC or 'CR'
			1)				1)

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

Telegram example:
Read a 60R data carrier
with BCC block check

Read the 60R data carrier.

The host sends 'S S' BCC (53 Hex)

The Handy Programmer acknowledges with <ACK> '0'
The host gives the start command <STX>
The Handy Programmer provides the datum from the data carrier '1 1' BCC (31 Hex)

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Telegram Structure and Examples
Reading / Writing over the serial interface

Initialize data carrier for a 60R system

This telegram is used to initialize a data carrier for the 60R system, i.e. the carrier is programmed with a Hex value.

Task	Data Flow	Com-mand	End	Acknow-ledge 2)	Start trans-mission	Datum	End	Acknow-ledge 2)
Write to a data carrier for a 60R system	from host system to BIS	**'U'**	BCC or 'CR'		<STX>	D	BCC or 'CR'	
	from BIS to host system			<ACK>'0' or <NAK> + Error-No.				<ACK>'0' or <NAK> + Error-No.
			1)				1)	

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

Telegram example:
Write to a 60R data carrier for initialization
with BCC block check

Write a Hex value to the 60R data carrier for initializatin.

The host sends 'U U' BCC (55 Hex)

The Handy Programmer acknowledges with <ACK> '0'
The host gives the start command and data <STX> '2 0' BCC (30 Hex)
The Handy Programmer acknowledges with <ACK> '0'

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Reading / Writing over the serial interface
Error Numbers

Error numbers

The BIS C-810 always outputs an error number. The meaning of these error numbers is indicated in the following table.

No.	Error Description	Effect	
2	Read error	Read telegram aborted, Handy Programmer goes into ground state.	
3	Read aborted, since the data carrier was removed	Handy Programmer goes into ground state.	
4	Write error	Write telegram aborted, Handy Programmer goes into ground state.	**CAUTION:** An aborted write could cause new data to be written to the data carrier which may be incomplete!
5	Writing aborted,since the data carrier was removed	Handy Programmer goes into ground state.	
6	Interface error	Handy Programmer goes into ground state. (parity or stop bit error)	
7	Telegram format error	Handy Programmer goes into ground state. Possible format errors: - Command is not 'L', 'P', 'Q', 'C', 'Z', 'S' or 'U'. - Start address or number of bytes exceed permissible range	
8	BCC error, the transmitted BCC is wrong	Telegram is aborted, Handy Programmer goes into ground state.	
E	CRC error	The CRC for the read data does not agree with the CRC on the data carrier.	

Reading / Writing over the serial interface
Read/Write Times

Read times
(Configuration: without CRC_16 check)

For double read and compare:

Data carrier with 32 byte blocks	
No. of bytes	Read time [ms]
from 0 to 31	110
for each additional 32 bytes add	120
from 0 to 255	= 950

Data carrier with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 63	220
for each additional 64 bytes add	230
from 0 to 2047	= 7350

Write times
(Configuration: without CRC_16 check)

Including readback and compare:

Data carrier with 32 byte blocks	
No. of bytes	Write time [ms]
from 0 to 31	110 + n * 10
for 32 bytes or more	y * 120 + n * 10

Data carrier with 64 byte blocks	
No. of bytes	Write time [ms]
from 0 to 63	220 + n * 10
for 64 bytes or more	y * 230 + n * 10

n = number of contiguous bytes to write
y = number of blocks to be written

Example:
Read 17 bytes starting at address 187. data carrier with 32 byte blocks. Blocks 5 and 6 have to be accessed, because start address 187 is in block 5 and end address 203 is in block 6.

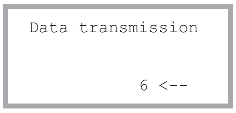
t = 2 * 120 + 17 * 10 = 410

The indicated times apply after the data carrier has been recognized. Otherwise an additional 45 ms must be added to allow for the energy field to be established until the data carrier is recognized. Times for sending the data over the interface must also be added.

Function Description BIS C-810 / PC
Working with files over the serial interface

Data transfer on the serial interface
4 3

Menu point 3 "Trans." can be used to transmit data over the serial interface from a PC or laptop to the Handy Programmer and back. It is also possible to administer the file. For this the BISHandy.exe program must be installed on the PC or laptop by running the file setup.exe. Both files are included on the supplied CD-ROM.



```
Data transmission

              6 <--
```

Data transmission

Select menu point 4 "File" in the Main Menu 1 and in the following menu select 3 "Trans." to activate the "Data transmission" function.

Data integrity on the serial interface

Each of the commands (D, G, T, K and E) is always terminated with a BCC block check. The Handy Programmer likewise terminates all transmitted characters with the BCC.

Telegram types with associated command (ASCII characters)

Specific telegrams exist for the individual data administration tasks in the Handy Programmer BIS C-810. They always begin with the command which corresponds to the telegram type:

'T'	Transmit file to the Handy Programmer
'G'	Get file from the Handy Programmer
'D'	Get file directory from the Handy Programmer
'E'	Delete a single file in the Handy Programmer
'K'	Delete all files in the Handy Programmer

Please note:
– Continuous querying on the interface is not permitted!
– The minimum wait time between two commands is 300 ms!
– With these telegrams only the BCC block check is used for data interity (regardless of the configuration).

Working with files over the serial interface
Telegram Structure and Examples

Send file from the PC to the Handy Programmer

This command is used to send a file stored in a PC or laptop to the Handy Programmer.

Task	Data Flow	Com-mand	File name 3)	Start address on the data carrier	No. of bytes on the data carrier	End	Acknow-ledge 2)	Start trans-mission	Data (from start address to start address + no. of bytes)	End	Acknow-ledge 2)
Send file to the Handy Programmer	from host system to BIS	**'T'**	'D7 ... D0' 'MUSTER_ _'	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '8 1 9 2'	BCC		<STX>	D1 D2 D3 ... Dn	BCC	
	from BIS to host system						<ACK>'0' or <NAK> + Error-No.				<ACK>'0' or <NAK> + Error-No.
			1)						1)		

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
3) If the file name is shorter than 8 characters, it must be filled out to 8 characters using spaces (20Hex).

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Working with files over the serial interface Telegram Structure and Examples

Read file from the Handy Programmer to the PC or laptop

This command is used to read out a file stored in the Handy Programmer.

Task	Data Flow	Com-mand	File name 3)	End	Acknow-ledge 2)	Start trans-mission	Start address on the data carrier	No. of bytes on the data carrier	Data (from start address to start address + no. of bytes)	End
Read file from the Handy Programmer	from host system to BIS	**'G'**	'D7 ... D0' 'MUSTER_ _'	BCC		<STX>				
	from BIS to host system				<ACK>'0' or <NAK> + Error-No.		A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '8 1 9 2'	D1 D2 D3 ... Dn	BCC
			1)				1)			

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
3) If the file name is shorter than 8 characters, it must be filled out to 8 characters using spaces (20Hex).

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Working with files over the serial interface Telegram Structure and Examples

Telegram example for 🗋 46:
Send file to the Handy Programmer
with BCC block check

Send file MUSTER1 to the Handy Programmer. Write 5 bytes to the data carrier starting with Address 27.

The host sends 'T MUSTER1␣ 0 0 2 7 0 0 0 5 M' BCC (4D Hex)

File name (filled out to 8 bytes) *)
Address of the first byte on the data carrier
Number of bytes

The Handy Programmer acknowledges with <ACK> '0'
The host gives the start command and data <STX> '12345 3' BCC (33 Hex)
The Handy Programmer acknowledges with <ACK> '0'

Telegram example for 🗋 47:
Read file from the Handy Programmer
with BCC block check

Read file MUSTER1 from the Handy Programmer. Read 5 bytes on the data carrier starting with Address 27.

The host sends 'G MUSTER1␣ ^' BCC (5E Hex)

File name (filled out to 8 bytes) *)

The Handy Programmer acknowledges with <ACK> '0'
The host sends <STX>
The Handy Programmer sends '0027000512345 1' BCC (33 Hex)

*) A space _, used for filling out the file name must also be accounted for in the BCC.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Working with files over the serial interface
Telegram Structure and Examples

Read file directory from Handy Programmer to PC / Laptop

This command is used to read the file directory stored in the Handy Programmer.

Task	Data Flow	Command	End	Acknowledge 2)	Start transmission	Data	End
Read directory from Handy Programmer	from host system to BIS	**'D'**	BCC		<STX>		
	from BIS to host system			<ACK>'0' or <NAK> + Error-No.		D1 D2 D3 ... Dn	BCC
			1)			1)	

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

Telegram example:
Read file directory from Handy Programmer
with BCC block check

Read the file directory from the Handy Programmer. Two files are stored in the Handy Programmer.

The host sends	'D D' BCC (44 Hex)
The Handy Programmer acknowledges with	<ACK> '0'
The host sends	<STX>
The Handy Programmer sends	'0002MUSTER1_00270005 MUSTER2_00350012 EOT'
	BCC (04 Hex)

*) A space _, used for filling out the file name must also be accounted for in the BCC.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Working with files over the serial interface
Telegram Structure and Examples

Delete a file in the Handy Programmer

This command is used to delete a file stored in the Handy Programmer.

Task	Data Flow	Command	Dateiname 3)	End	Acknowledge 2)
Delete a file in the Handy Programmer	from host system to BIS	**'E'**	'D7 ... D0' 'MUSTER_ _'	BCC	
	from BIS to host system				<ACK>'0' or <NAK> + Error-No.
			1)		

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
3) If the file name is shorter than 8 characters, it must be filled out to 8 characters using spaces (20Hex).

Telegram example:
Delete file in Handy Programmer
with BCC block check

Delete file MUSTER1 from the directory in the Handy Programmer.

The host sends	'E MUSTER1␣ \' BCC (5C Hex)
File name (filled out to 8 bytes) *)	
The Handy Programmer acknowledges with	<ACK> '0'

*) A space _, used for filling out the file name must also be accounted for in the BCC.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Working with files over the serial interface
Telegram Structure and Examples

Delete all files in the Handy Programmer

This command is used to delete all files in the Handy Programmer.

Task	Data Flow	Command	End	Acknowledge 2)
Delete all files in Handy Programmer	from host system to BIS	'K'	BCC	
	from BIS to host system			<ACK>'0' or <NAK> + Error-No.
			1)	

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
3) If the file name is shorter than 8 characters, it must be filled out with spaces (20Hex).

Telegram example: ***Delete all files in the Handy Programmer*** *with BCC block check*

Delete all files in the Handy Programmer

The host sends 'K K' BCC (4B Hex)

The Handy Programmer acknowledges with <ACK> '0'

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Working with files over the serial interface
Error Numbers

Error numbers

The BIS C-810 always outputs an error number. The meaning of these error numbers is indicated in the following table.

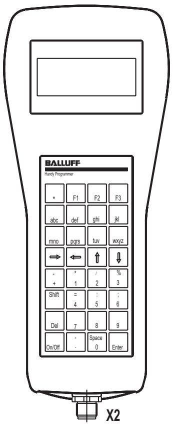
Error No.	Error description	Cause
0 = 30 Hex	No error	Only in conjunction with <ACK>
Q = 51 Hex	Timeout	Character delay time of 1 s exceedced. Interface proble
R = 52 Hex	Entry error	Specified file not found

S = 53 Hex
T = 54 Hex
	Memory error	End of Handy Programmer memory reached

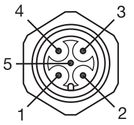
V = 56 Hex	Interface error	Incorrect port settings (baud rate, parity, data bit or stop error)
W = 57 Hex	Protocol format error	Protocol format not maintained
X = 58 Hex	BCC is incorrect	Incorrect BCC returned

BIS C-810
Interface Information

Serial interface RS 232



X2, serial interface RS 232



Pin	Function
1	n.c.
2	TxD
3	GND
4	RxD
5	n.c.

n.c. = do not connect

BIS C-810
Battery charging

Battery charging

There are two ways to charge the battery in the Handy Programmer:

Without removing: Insert the 2.5 mm jack on the BIS C-701/A *) charger into the charging socket on the bottom of the Handy Programmer.

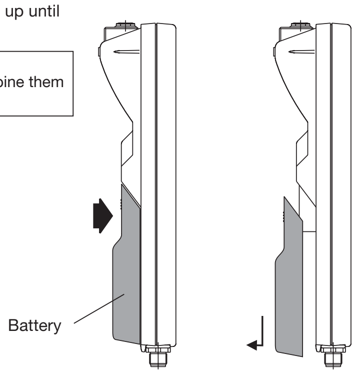
Removed: For this the battery is removed (see below) and inserted into the BIS C-702-A*) charging holder.

Replace battery

To remove the battery from the Handy Programmer, press the latch in and slide the battery down slightly until the guides release the battery and it can be removed.

After charging, reinsert the battery and slide it up until it latches.

Please properly dispose of used batteries.
Never throw them into bodies of water or combine them with normal refuse.



Removing the battery

*) See Accessories on 🗋 58

BIS C-810 Technical Data

Housing	Housing material Dimensions Weight incl. battery	ABS approx. 97 × 232 × 55 mm approx. 420 g
	Keypad	32 keys, 4 × 8 mm, alphanumeric and special characters
	Display	LCD-Display, 80 characters / 4 lines
Connections	RS 232 serial port	5-pin round connector
	Read/Write Head	6.3 mm jack suitable for all C-85_ with 6.3 mm jack plug
	Charging socket	2.5 mm jack designed for BIS C-701-A
Enclosure rating	Enclosure rating	IP 40 (when connected)
Temperature range	Ambient temperature	0 °C to +50 °C

BIS C-810 Technical Data

Electrical data	Power supply Capacity max. charge voltage Charging current Charging time	2.4 V NiMH rechargeable battery 1650 mAh 5–10 V, internally limited approx. 400 mA approx. 4 h
Current draw	Current draw, normal operation Current draw, reading/writing	approx. 65 mA approx. 275 mA

With the CE Mark we affirm that our products are in accordance with the requirements of the EU (European Union) Guideline

89/336/EEC (EMC Guideline)

and the EMC Law. It has been verified in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, that Balluff products meet the EMC requirements of the Harmonized Standard

EN 61000-6-4 (Emission), EN 61000-6-2 (Noise Immunity)

BIS C-810
Ordering Information

Ordering code

BIS C-810-0-003

Balluff Identification System

Series C

Hardware type
810 = Plug-in (6.3 mm jack) for BIS C-85_ read/write head,
with keypad and display, for connecting to PC
or laptop with 9-pin SUB-D connector (serial)

Interface
0 = RS 232 (V.24 Interface)

Software type
003 = Balluff software for BIS C-810-...

BIS C-810
Ordering Information

Accessories
(optional,
not included)

Article	**Ordering code**
Read/write head with 6.3 mm jack plug suitable for BIS C-100-_ _/A or BIS C-122-_ _/L data carriers	BIS C-851
Read/write head with 6.3 mm jack plug suitable for BIS C-117-_ _/L, BIS C-117-_ _/A, BIS C-128-_ _/L or BIS C-130-_ _/L data carriers	BIS C-852
Read/write head with 6.3 mm jack plug suitable for BIS C-150-_ _/A data carrier	BIS C-853
Adapter for all BIS C-3_ _ read/write heads, except BIS C-350 and BIS C-352	BIS C-850
Charger	BIS C-701-A
Charge holder	BIS C-702-A
Interface cable for connecting the BIS C-810 Handy Programmer to a 9-pin SUB-D COM port on a laptop or PC	BIS C-522

Appendix, ASCII Table

Deci-mal	Hex	Control Code	ASCII	Deci-mal	Hex	Control Code	ASCII	Deci-mal	Hex	ASCII	Deci-mal	Hex	ASCII	Deci-mal	Hex	ASCII	Deci-mal	Hex	ASCII
0	00	Ctrl @	NUL	22	16	Ctrl V	SYN	44	2C	,	65	41	A	86	56	V	107	6B	k
1	01	Ctrl A	SOH	23	17	Ctrl W	ETB	45	2D	-	66	42	B	87	57	W	108	6C	l
2	02	Ctrl B	STX	24	18	Ctrl X	CAN	46	2E	.	67	43	C	88	58	X	109	6D	m
3	03	Ctrl C	ETX	25	19	Ctrl Y	EM	47	2F	/	68	44	D	89	59	Y	110	6E	n
4	04	Ctrl D	EOT	26	1A	Ctrl Z	SUB	48	30	0	69	45	E	90	5A	Z	111	6F	o
5	05	Ctrl E	ENQ	27	1B	Ctrl [	ESC	49	31	1	70	46	F	91	5B	[	112	70	p
6	06	Ctrl F	ACK	28	1C	Ctrl \	FS	50	32	2	71	47	G	92	5C	\	113	71	q
7	07	Ctrl G	BEL	29	1D	Ctrl]	GS	51	33	3	72	48	H	93	5D	]	114	72	r
8	08	Ctrl H	BS	30	1E	Ctrl ^	RS	52	34	4	73	49	I	94	5E	^	115	73	s
9	09	Ctrl I	HT	31	1F	Ctrl _	US	53	35	5	74	4A	J	95	5F	_	116	74	t
10	0A	Ctrl J	LF	32	20		SP	54	36	6	75	4B	K	96	60	`	117	75	u
11	0B	Ctrl K	VT	33	21		!	55	37	7	76	4C	L	97	61	a	118	76	v
12	0C	Ctrl L	FF	34	22		"	56	38	8	77	4D	M	98	62	b	119	77	w
13	0D	Ctrl M	CR	35	23		#	57	39	9	78	4E	N	99	63	c	120	78	x
14	0E	Ctrl N	SO	36	24		$	58	3A	:	79	4F	O	100	64	d	121	79	y
15	0F	Ctrl O	SI	37	25		%	59	3B	;	80	50	P	101	65	e	122	7A	z
16	10	Ctrl P	DLE	38	26		&	60	3C	<	81	51	Q	102	66	f	123	7B	{
17	11	Ctrl Q	DC1	39	27		'	61	3D	=	82	52	R	103	67	g	124	7C	|
18	12	Ctrl R	DC2	40	28		(	62	3E	>	83	53	S	104	68	h	125	7D	}
19	13	Ctrl S	DC3	41	29		)	63	3F	?	84	54	T	105	69	i	126	7E	~
20	14	Ctrl T	DC4	42	2A		*	64	40	@	85	55	U	106	6A	j	127	7F	DEL
21	15	Ctrl U	NAK	43	2B		+												